QMM_Assignment4

Ram

10/25/2021

Q2)

1) What is the minimum cost of providing oil to the refineries? Which wells are used to capacity in the optimal schedule?

Supply is 276 TBD and the demand is 274 TBD. So, the demand is not equal to supply. So, we create a dummy variable in the demand side of 2 TBD to make sure that the demand is equal to the supply.

As given in the problem, the minimum objective function is formulated below:

Zmin = 1.52 X1A + 1.60 X1B + 1.40 X1C + 1.70 X2A + 1.63 X2B + 1.55 X2C + 1.45 X3A + 1.57 X3B + 1.30 X3C + 5.15 XA1 + 5.12 XB1 + 5.32 XC1 + 5.69 XA2 + 5.47 XB2 + 6.16 XC2 + 6.13 XA3 + 6.05 XB3 + 6.25 XC3 + 5.63 XA4 + 6.12 XB4 + 6.17 XC4 + 5.80 XA5 + 5.71 XB5 + 5.87 XC5 + 0 X A6 + 0 XB6 + 0 XC6

Constraints:

Supply Constraints X1A +X1B + X1C = 93 X2A + X2B + X2C = 88 X3A + X3B + X3C = 95

Demand Constraints XA1 + XB1 + XC1 = 30 XA2 + XB2 + XC2 = 57 XA3 + XB3 + XC3 = 48 XA4 + XB4 + XC4 = 91 XA5 + XB5 + XC5 = 48 XA6 + XB6 + XC6 = 2

Constraints from pumps to refinery

X1A + X2A + X3A = XA1 + XA2 + XA3 + XA4 + XA5 + XA6 X1B + X2B + X3B = XB1 + XB2 + XB3 + XB4 + XB5 + XB6 X1C + X2C + X3C = XC1 + XC2 + XC3+ XC4 + XC5 + XC6 Where, Xij >= 0: i(pumps) = (A, B, C), j= 1,2,3(wells), 1:6(refineries)

Using Ipsolve the optimal solution is 1966.68.

Well 3 has used to the capacity in the optimal schedule.

```
library(lpSolveAPI)
lprec<-make.lp(0,27)
lp.control(lprec,sense='min')

## $anti.degen
## [1] "fixedvars" "stalling"
##
## $basis.crash
## [1] "none"
##
```

```
## $bb.depthlimit
## [1] -50
##
## $bb.floorfirst
## [1] "automatic"
##
## $bb.rule
## [1] "pseudononint" "greedy"       "dynamic"      "rcostfixing"
##
## $break.at.first
## [1] FALSE
##
## $break.at.value
## [1] -1e+30
##
## $epsilon
##       epsb       epsd      epsel     epsint epsperturb   epspivot
##      1e-10      1e-09      1e-12      1e-07      1e-05      2e-07
##
## $improve
## [1] "dualfeas" "thetagap"
##
## $infinite
## [1] 1e+30
##
## $maxpivot
## [1] 250
##
## $mip.gap
## absolute relative
##    1e-11    1e-11
##
## $negrange
## [1] -1e+06
##
## $obj.in.basis
## [1] TRUE
##
## $pivoting
## [1] "devex"    "adaptive"
##
## $presolve
## [1] "none"
##
## $scalelimit
## [1] 5
##
## $scaling
## [1] "geometric"   "equilibrate" "integers"
##
```

```
## $sense
## [1] "minimize"
##
## $simplextype
## [1] "dual"   "primal"
##
## $timeout
## [1] 0
##
## $verbose
## [1] "neutral"
```

```
set.objfn(lprec,c(1.52,1.60,1.40,1.70,1.63,1.55,1.45,1.57,1.30,5.15,5.12,5.32
,5.69,5.47,6.16,6.13,6.05,6.25,5.63,6.12,6.17,5.80,5.71,5.87,0,0,0))
add.constraint(lprec,c(1,1,1),"=",93,indices = c(1,2,3))
add.constraint(lprec,c(1,1,1),"=",88,indices = c(4,5,6))
add.constraint(lprec,c(1,1,1),"=",95,indices = c(7,8,9))
add.constraint(lprec,c(1,1,1),"=",30,indices = c(10,11,12))
add.constraint(lprec,c(1,1,1),"=",57,indices = c(13,14,15))
add.constraint(lprec,c(1,1,1),"=",48,indices = c(16,17,18))
add.constraint(lprec,c(1,1,1),"=",91,indices = c(19,20,21))
add.constraint(lprec,c(1,1,1),"=",48,indices = c(22,23,24))
add.constraint(lprec,c(1,1,1),"=",2,indices = c(25,26,27))
add.constraint(lprec,c(rep(1,3),rep(-
1,6)),"=",0,indices=c(1,4,7,10,13,16,19,22,25))
add.constraint(lprec,c(rep(1,3),rep(-
1,6)),"=",0,indices=c(2,5,8,11,14,17,20,23,26))
add.constraint(lprec,c(rep(1,3),rep(-
1,6)),"=",0,indices=c(3,6,9,12,15,18,21,24,27))
solve(lprec)
```

```
## [1] 0
```

```
get.objective(lprec)
```

```
## [1] 1966.68
```

```
get.constraints(lprec)
```

```
##  [1] 93 88 95 30 57 48 91 48  2  0  0  0
```

```
get.variables(lprec)
```

```
##  [1] 93  0  0  0 88  0 28  0 67 30  0  0  0 57  0  0 31 17 91  0  0  0  0
48  0
## [26]  0  2
```